

What is claimed is:

1 1. A method of processing access requests for a direct access storage device
2 (DASD), each access request associated with a requester and a position on the DASD,
3 the method comprising:

4 (a) sorting at least a subset of a plurality of access requests directed to
5 the DASD based upon the requesters associated therewith to generate a first
6 ordered set of access requests;

7 (b) sorting at least a subset of the access requests in the first ordered
8 set of access requests based upon the positions associated therewith to
9 generate a second ordered set of access requests; and

10 (c) issuing each of the access requests in the second ordered set of
11 access requests in sequence to the DASD.

1 2. The method of claim 1, further comprising moving access requests between
2 a first queue and a second queue, wherein sorting the access requests to generate the
3 first ordered set of access requests includes receiving inbound access requests into the
4 first queue, and sorting the access requests received into the first queue based upon
5 the requesters associated therewith, wherein sorting the access requests in the first
6 ordered set includes sorting access requests moved into the second queue based upon
7 the positions associated therewith, and wherein issuing the access requests in the
8 second ordered set includes removing issued access requests from the second queue.

1 3. The method of claim 2, wherein sorting the access requests received into
2 the first queue includes, for each inbound access request:

3 (a) determining whether another access request in the first queue is
4 associated with the same requester as the inbound access request;

5 (b) if so, storing the inbound access request after a last access request
6 associated with the same requester as the inbound access request; and

7 (c) if not, arranging the inbound access request within the first queue
8 based upon a requester identifier associated therewith.

1 4. The method of claim 3, wherein moving access requests between the first
2 queue and the second queue is performed in response to determining that the second
3 queue is empty.

1 5. The method of claim 4, wherein moving access requests between the first
2 queue and the second queue includes, for each requester associated with an access
3 request stored in the first queue, moving at least one access request associated with
4 such requester into the second queue.

1 6. The method of claim 2, wherein moving access requests between the first
2 queue and the second queue further includes moving a batch of access requests
3 between the first queue and second queue, and wherein sorting the access requests
4 moved into the second queue includes reversing a sort order for each successive batch
5 of access requests.

1 7. The method of claim 1, wherein sorting the access requests in the first
2 ordered set includes alternately sorting access requests in ascending and descending
3 order.

1 8. The method of claim 1, wherein each access request is associated with one
2 of a plurality of requesters in a computer coupled to the DASD, with each requester
3 being a computer task executing on the computer.

1 9. A method of processing access requests for a direct access storage device
2 (DASD), each access request associated with a requester and a position on the DASD,
3 the method comprising:

- 4 (a) sorting a plurality of access requests directed to the DASD based
5 upon both the requesters and the positions associated therewith; and
6 (b) issuing the sorted access requests to the DASD.

1 10. The method of claim 9, wherein sorting the plurality of access requests
2 includes:

3 (a) sorting at least a subset of a plurality of access requests based upon
4 the requesters associated therewith to generate a first ordered set of access
5 requests; and

6 (b) sorting at least a subset of the access requests in the first ordered
7 set of access requests based upon the positions associated therewith to
8 generate a second ordered set of access requests, wherein issuing the sorted
9 access requests includes issuing each of the access requests in the second
10 ordered set of access requests in sequence to the DASD.

1 11. The method of claim 9, wherein sorting the plurality of access requests
2 includes:

3 (a) sorting at least a subset of a plurality of access requests using one
4 of a fair scheduling algorithm and an elevator scheduling algorithm to generate
5 a first ordered set of access requests; and

6 (b) sorting at least a subset of the access requests in the first ordered
7 set of access requests using the other of the fair scheduling algorithm and the
8 elevator scheduling algorithm.

1 12. A method of processing access requests for a direct access storage device
2 (DASD), each access request associated with a requester and a position on the DASD,
3 the method comprising:

4 (a) receiving incoming access requests into a first queue that is sorted
5 based upon the requester associated with each access request stored in the first
6 queue;

7 (b) moving at least one access request from the first queue to a second
8 queue that is sorted based upon the position associated with each access
9 request stored in the second queue; and

10 (c) sequentially issuing access requests from the second queue to the
11 DASD.

664021-123456

1 13. An apparatus for use in processing access requests for a direct access
2 storage device (DASD), each access request associated with a requester and a position
3 on the DASD, the apparatus comprising:

4 (a) a memory; and

5 (b) a program, resident in the memory, the program configured to sort
6 a plurality of access requests directed to the DASD based upon both the
7 requesters and the positions associated therewith, and issue the sorted access
8 requests to the DASD.

1 14. The apparatus of claim 13, wherein the program is configured to sort the
2 plurality of access requests by sorting at least a subset of a plurality of access requests
3 using one of a fair scheduling algorithm and an elevator scheduling algorithm to
4 generate a first ordered set of access requests; and sorting at least a subset of the
5 access requests in the first ordered set of access requests using the other of the fair
6 scheduling algorithm and the elevator scheduling algorithm.

1 15. The apparatus of claim 13, wherein the program is configured to sort the
2 plurality of access requests by sorting at least a subset of a plurality of access requests
3 based upon the requesters associated therewith to generate a first ordered set of access
4 requests, and sorting at least a subset of the access requests in the first ordered set of
5 access requests based upon the positions associated therewith to generate a second
6 ordered set of access requests, and wherein the program is configured to issue the
7 sorted access requests by issuing each of the access requests in the second ordered set
8 of access requests in sequence to the DASD.

1 16. The apparatus of claim 15, wherein the program is further configured to:

2 (a) move access requests between a first queue and a second queue;

3 (b) sort the access requests to generate the first ordered set of access
4 requests by receiving inbound access requests into the first queue and sorting
5 the access requests received into the first queue based upon the requesters
6 associated therewith;

7 (c) sort the access requests in the first ordered set by sorting access
8 requests moved into the second queue based upon the positions associated
9 therewith; and

10 (d) issue the access requests in the second ordered set by removing
11 issued access requests from the second queue.

1 17. The apparatus of claim 16, wherein the program is configured to sort the
2 access requests received into the first queue by, for each inbound access request:

3 (a) determining whether another access request in the first queue is
4 associated with the same requester as the inbound access request;

5 (b) if so, storing the inbound access request after a last access request
6 associated with the same requester as the inbound access request; and

7 (c) if not, arranging the inbound access request within the first queue
8 based upon a requester identifier associated therewith.

1 18. The apparatus of claim 17, wherein the program is configured to move
2 access requests between the first queue and the second queue in response to
3 determining that the second queue is empty.

1 19. The apparatus of claim 18, wherein the program is configured to move
2 access requests between the first queue and the second queue by, for each requester
3 associated with an access request stored in the first queue, moving at least one access
4 request associated with such requester into the second queue.

1 20. The apparatus of claim 16, wherein the program is configured to move
2 access requests between the first queue and the second queue by moving a batch of
3 access requests between the first queue and second queue, and wherein the program is
4 configured to sort the access requests moved into the second queue by reversing a sort
5 order for each successive batch of access requests.

1 21. The apparatus of claim 15, wherein the program is configured to sort the
2 access requests in the first ordered set by alternately sorting access requests in
3 ascending and descending order.

1 22. The apparatus of claim 15, further comprising a plurality of requester
2 computer tasks resident in the memory, wherein each access request is associated with
3 one of the plurality of requester computer tasks.

00456211-120799

1 23. An apparatus for use in processing access requests for a direct access
2 storage device (DASD), each access request associated with a requester and a position
3 on the DASD, the apparatus comprising:

4 (a) a first queue configured to receive incoming access requests, the
5 access requests in the first queue sorted based upon the requesters associated
6 therewith;

7 (b) a second queue configured to issue outgoing access requests to the
8 DASD, the access requests in the second queue sorted based upon the
9 positions associated therewith; and

10 (c) control logic coupled to the first and second queues and configured
11 to selectively move access requests from the first queue to the second queue.

1 24. The apparatus of claim 23, further comprising a memory; and a processor
2 coupled to the memory, wherein the control logic comprises request processing
3 program code resident in the memory.

1 25. The apparatus of claim 24, further comprising an operating system
2 resident in the memory, wherein the request processing program code and the first and
3 second queues are resident in the operating system.

1 26. The apparatus of claim 25, wherein the operating system includes a
2 DASD hardware driver that interfaces the apparatus with the DASD, and wherein the
3 request processing program code and the first and second queues are resident in the
4 DASD hardware driver.

1 27. The apparatus of claim 24, wherein the first and second queues are
2 resident in the memory.

1 28. A program product, comprising:

2 (a) a program for use in processing access requests for a direct access
3 storage device (DASD), each access request associated with a requester and a
4 position on the DASD, the program configured to sort a plurality of access
5 requests directed to the DASD based upon both the requesters and the
6 positions associated therewith, and issue the sorted access requests to the
7 DASD; and

8 (b) a signal bearing medium bearing the program.

1 29. The program product of claim 28, wherein the signal bearing medium
2 includes at least one of a recordable medium and a transmission type medium.

0045641-120789